ContraDoc: Understanding Self-Contradictions in Documents with Large Language Models

Anonymous NAACL submission

Abstract

In recent times, large language models (LLMs) have shown impressive performance on various document-level tasks such as document classification, summarization, and question-answering. However, research on understanding their capabilities on the task of self-contradictions in long documents has been very limited. In this work, we introduce ContraDoc, the first human-annotated dataset to study self-contradictions in long documents across multiple domains, varying document lengths, self-contradictions types, and scope. We then analyze the current capabilities of four state-of-the-art open-source and commercially available LLMs: GPT3.5, GPT4, PaLM2, and LLaMAv2 on this dataset. While GPT4 performs the best and can outperform humans on this task, we find that it is still unreliable and struggles with self-contradictions that require more nuance and context. We release the dataset and all the code associated with the experiments.

Document Type: News Article

.... So high, that it is taking five surgeons, a covey of physician assistants, nurses and anesthesiologists, and more than 40 support staff to perform surgeries on 12 people. They are extracting six kidneys from donors and implanting them into six recipients..... In late March, the medical center is planning to hold a reception for all 10 patients. Here's how the super swap works, according to California Pacific Medical Center. ....

Scope of Self-Contradiction: Global

Type of Self-Contradiction: Numeric, Content

Figure 1: Example of a self-contradictory document from ContraDoc. The highlighted parts in green show the evidence for the self-contradiction. Additionally, information about the scope and type of the contradiction is also present.

1 Introduction

Detecting contradictions in texts has long been pivotal in natural language understanding(NLU), with most of the works falling under the umbrella of natural language inference(NLI)(Harabagiu et al., 2006; Dagan et al., 2005; de Marneffe et al., 2008). Detecting contradictions is often regarded as determining the relation between a hypothesis and a piece of premise. However, understanding contradictions when they occur within the confines of a single text (self-contradictions), and furthermore, doing so holistically at the document-level, is still under-explored. A text is considered self-contradictory when it contains multiple ideas or statements that inherently conflict. This could manifest in multiple different ways, such as the existence of logical paradoxes, antithetical assertions, or inconsistent descriptions. Figure 1 shows an example of self-contradiction in a document. The highlighted two sentences provide contradictory information about the number of patients, thus resulting in a self-contradictory document.

Psychological research (Graesser and McMahen, 1993; Otero and Kintsch, 1992) indicates that humans struggle to identify contradictions in unfamiliar, informative texts, particularly when contradictions are widely separated in long documents, underscoring the need for automated text analysis tools to tackle this challenge.

Previous research on document-level contradictions either focused on sentence-document pair NLI(Yin et al., 2021a; Schuster et al., 2022a) or has been restricted to a single type of document(Hsu et al., 2021). Hsu et al. (2021) defined self-contradiction detection as a binary classification task, proving inadequate for accurately evaluating and locating self-contradictions within texts.

Therefore, we propose a new document-level self-contradictory dataset ContraDoc with the following characteristics:

- documents are from different sources and of different lengths.
- The documents and the highlighted self-contradictions within are verified by human annotators.
- It contains a variety of self-contradictions,

with each contradiction tagged with information such as its type and scope by human annotators.
- The resulting self-contradictory documents are contextually fluent, thus, keeping the document coherent and plausible.

To create CONTRADOC, we utilize a human-machine collaborative framework. We first use LLMs and NLP pipelines to automatically create and introduce self-contradiction into a consistent document. Then, human annotators verify and label attributes for the self-contradictory documents, ensuring the quality and utility of our dataset.

The advent of large language models (LLMs) pre-trained on extensive context lengths (Brown et al., 2020a; Chowdhery et al., 2022); have shown promising results over various document-level tasks spanning document classification(Sun et al., 2023), document summarization(Zhang et al., 2023), document-level question answering(Singhal et al., 2023), and document-level machine translation(Wang et al., 2023). To investigate how well can large language models detect self-contradiction in documents, we evaluate state-of-the-art, open-source and commercially available LLMs: GPT3.5(OpenAI, 2022), GPT4(OpenAI, 2023), PaLM2(Anil et al., 2023), and LLaMAv2(Touvron et al., 2023) on CONTRADOC.

We design three evaluation tasks and corresponding metrics to assess LLMs' performance in a zero-shot setting. In our experiments, we find that even SOTA models cannot achieve applicable performance. We did a thorough study on the effects of different aspects of documents and self-contradictions.

In summary, this paper makes the following contributions:

- We propose a human-annotated dataset consisting of self-contradictory documents across varying document domains and lengths and self-contradiction types and scope, being the first work to touch on those aspects.
- We propose three evaluation tasks and metrics to evaluate the performance of models on detecting self-contradictions in text. They evaluate not just binary judgment but also the models' ability to pinpoint contradictions within the documents.
- We conduct an extensive analysis of four of the best-performing LLMs (open-source and commercially available) and provide insights into their capabilities of long-form reasoning, focused on self-contradiction detection in documents.

2 Related Work

2.1 Detecting Contradictions in Text

The problem of detecting contradictory statements in texts has been long explored in NLP literature (Condoravdi et al., 2003; Harabagiu et al., 2006), mainly as a text classification or textual entailment task. Most prior work has studied contradictions under the Natural Language Inference (NLI) framework of evaluating contradictory pairs of sentences, namely, as Recognizing Textual Entailment (RTE) tasks (Dagan et al., 2005; Bowman et al., 2015). Contradiction detection has also been explored in dialogue Nie et al. (2021); Zheng et al. (2022); Jin et al. (2022), question answering systems (Fortier-Dubois and Rosati, 2023).

More recently, a fair amount of NLI research has focused on long-document reasoning, extending beyond sentence-level granularity to document-level,(Yin et al., 2021b; Schuster et al., 2022b; Mathur et al., 2022). However, these works differ from ours as they either frame the tasks as NLI, do not focus on investigating the capabilities of LLMs, or do not focus on self-contradictions.

2.2 Understanding Self-Contradictions

Despite the extensive amount of research into studying contradictions, there has been a very limited amount of work that has focused on self-contradictions in long documents. The closest work to ours is Hsu et al. (2021) on Wikipedia-based contradiction detection, where they curated a dataset based on the "Self-contradictory" template on Wikipedia and used a pairwise model to detect it. CONTRADOCdataset significantly differs from their proposed dataset in the variety of document types, contradiction types and additional annotations it contains. Mündler et al. (2023) refine LLM-generated texts to eliminate contradictions, both relevant yet distinct from our comprehensive, domain-inclusive approach focusing on holistic document analysis with LLMs.

3 CONTRADOC Dataset

CONTRADOC contains 449 self-contradictory (referred to as CONTRADOC-POS) and 442 non-contradictory documents (referred to as

CONTRADOC-NEG). Non-contradictory documents are defined as documents that do not contain any self-contradictions and are considered negative examples for the task. We include them in our dataset to evaluate if the models can identify the documents that do not contain any self-contradictions sampled from the same source of contradictory documents. Furthermore, the documents in CONTRADOC cover three domains, vary in length and scope of dependencies, and contain different types of contradictions. This allows us to see how these variations affect the performance of the LLMs. In the development of our dataset, we leverage a human-machine collaborative framework, where human experts evaluate and verify machine-generated self-contradictions, ensuring the created data is both rich and reliable. We only use documents written in English in this work.

3.1 Dataset Statistics

The overall statistics for the 449 documents in CONTRADOC-POSare shown later in this paper in Table 5. The distribution of non-contradictory documents in CONTRADOC-NEG is similar to CONTRADOC-POS .

The different attributes of our dataset pertaining to self-contradiction types, document, and context lengths, and the research questions used to study them are outlined below.

RQ1: Are self-contradictions harder to detect in some domains for LLMs? To create CONTRADOC, we construct a document corpus from three domains to test the performance in various contexts. We use CNN-DailyMail dataset (Hermann et al., 2015) for news articles, NarrativeQA (Kočiskỳ et al., 2018) for stories, and WikiText (Merity et al., 2016) for Wikipedia documents (details in Appendix A).

RQ2: Are self-contradiction harder to detect in longer documents for LLMs? Documents in CONTRADOCrange from 100 tokens to 2200 tokens helping us study both longer and shorter documents. Table 5 shows the detailed breakdown of our dataset with respect to document lengths (in tokens).

RQ3: Are self-contradictions present farther away in a document more difficult to detect for LLMs? The instances where contradictions are present within a sentence are labeled as *intra*, whereas the instances where the contradictory statements are present four sentences or less apart are labeled *local*, and finally, the instances where the contradictions are present more than four sentences apart are labeled *global*. Our dataset contains 73, 220, and 155 documents with intra, local, and global contradictions.

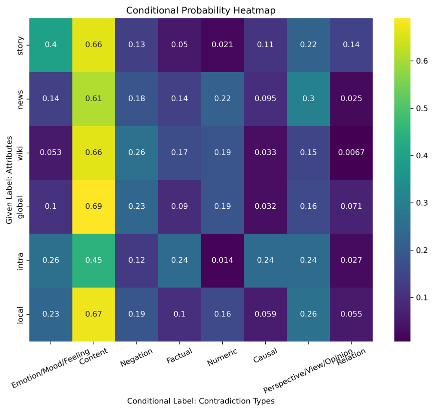

Figure 2: Label dependencies, shown with conditional probabilities. Each cell is the occurrence probability of the x-axis label, given the presence of the y-axis label.

RQ4: Are some types of self-contradictions harder to detect than others for LLMs? Each document in CONTRADOCis tagged with one or multiple of the following eight types of self-contradictions: Negation, Numeric, Content, Perspective/View/Opinion, Emotion/Mood/Feeling, Factual, Relation, and Causal. A more comprehensive overview is presented in Appendix C.

The labeled attributes in our dataset are not independent of each other. We illustrate the conditional probabilities over the contradiction types and other properties in Figure 2 to show the dependencies between them. For the self-contradiction type, "Content" is the most common type as it often co-occurs with other types like "Negation", "Numeric" or "Factual". We notice that 40% of story documents contain "Emotion/Mood/Feeling" self-contradiction while this number is only "14%" and "5.3%" for news and wiki, showing that the distributions of types of self-contradictions vary a lot amongst different types of documents. The dependency effect should be taken into consideration as we analyze the more fine-grained performance on different labels in experiments (more in Section 4.4).

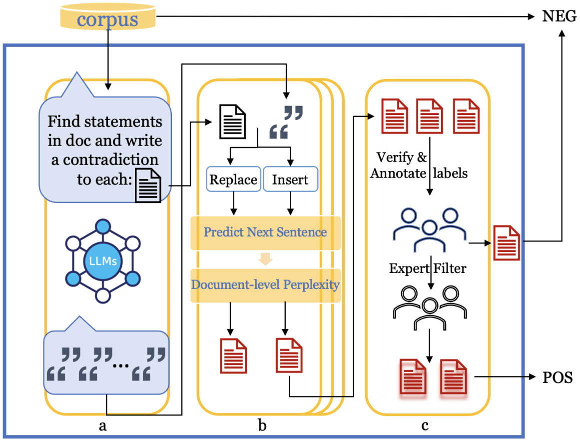

Figure 3: Dataset Creation Pipeline. a) Contradictory Statements Generation using LLMs; b) Self-Contradictory Document Creation; c) Human verification and Tagging.

3.2 Dataset Creation Method

While LLMs are used more and more in data labeling and dataset creation (Ding et al., 2023; Wang et al., 2021), Pangakis et al. (2023) argues that the data annotated by generative AI requires human verification. Thus, we utilize a human-machine collaborative framework to create our dataset. We first automatically create and introduce self-contradictions into a document. Then, we ask human annotators to verify and label attributes for the contradictory documents. The data creation process is systematically organized into three primary components: a) Contradictory Statements Generation; b) Self-Contradictory Document Creation; c) Human Verification and Tagging. Figure 3 provides an overview of the dataset creation process.

3.2.1 Contradictory Statements Generation Using LLM

Given a document d, we process it through an LLM (GPT-4-0314 in our case) to generate contradictory statements by asking it to identify k statements $st_1, st_2, \cdots, st_k$ in the document and generate a contradictory statement to each of the k statements, yielding k contradictions correspondingly: $c_1, c_2, \cdots, c_k$. More specifically, we provide few-shot examples of contradictory statements of different types, guiding the LLM to identify and generate more diverse statements.

In practice, the model tends to edit only a few words in the statement unless explicitly asked otherwise. To make contradictory statements sound natural, we also ask it to rephrase it using a different wording $c_1^{'}, c_2^{'}, \cdots, c_k^{'}$. Thus for a single document provided, LLM generates k triplets: $(st_i, c_i, c_i^{'})$

3.2.2 Self-Contradictory Document Creation

Upon obtaining k of $(st_i, c_i, c_i^{'})$ triplets, we modify the source document by either *inserting* the contradictory statement c_i or $c_i^{'}$ in the document or *replacing* the original statement st_i with c_i or $c_i^{'}$, forming a candidate set of potentially contradictory documents $\hat{D}_i = \{\hat{d}_i(ins - c_i), \hat{d}_i(ins - c_i^{'}), \hat{d}_i(rep - c_i), \hat{d}_i(rep - c_i^{'})\}$. This is driven by two assumptions: 1) Introducing contradictory facts separately may render the document self-contradictory. 2) Directly substituting statements with contradictory versions might induce contextual inconsistency.

To maintain document fluency while introducing contradiction, we apply the following metrics to filter in self-contradictory documents from the candidate set:

- **Global Fluency**: We measure document-level perplexity and ensure that it does not exceed a defined threshold, T, post-editing.

$$ppl(d) = exp(1/n) * \sum_{j=1}^{n}(log(P(w_j)))$$

$$ppl(\hat{d}_i) - ppl(d) <= T$$

  where n is the total number of tokens in document d and $P(w_j)$ is the probability to predict token w_j. In practice, we set $T = 0.01$ to 0.03 for different types and lengths of documents.
- **Local Fluency**: We employ BERT's "Next Sentence Prediction(NSP)" task (Devlin et al., 2019) to validate the contextual coherence of the modified sentences. After placing the modified sentence in c_i or $c_i^{'}$ at position j th, we accept such edit if: NSP(s_{j-1}, s_j) and NSP(s_j, s_{j+1}) are both True.

If multiple contradictory documents in $\hat{D}_i$ meet the mentioned constraints, we accept the one with the lowest global perplexity to maintain diversity in self-contradictions.

3.2.3 Human Verification and Tagging

An additional human annotation layer was integrated to validate the automated modifications, ensuring the resultant documents were both natural and genuinely contradictory. We highlight the original statement as well as the introduced self-contradiction in the document as Figure 1 for annotators[1] to verify the validity of document-level self-contradiction as well as tagging labels for self-contradiction type and scope of self-contradiction(intra, local, global as in Section 3.1). The questions can be found in D.

Each modified document was evaluated by two annotators, establishing consensus on the self-contradiction and document validity. Examples are filtered if both annotators verify that the modification makes a valid document-level self-contradiction. When annotators disagree, we select "closer" option for self-contradiction scope while joining different self-contradiction types.

To verify the annotation quality, we run another expert filter by the authors of this work to verify controversial cases marked by annotators. Regarding the self-contradiction injection method, the final CONTRADOC contains 271 documents created by contradictory statement replacing and 178 documents created by contradictory statement inserting.

3.2.4 Negative Examples

We consider the documents without self-contradictions as negative examples in our experiments. While the documents from our source domain can naturally serve as negative examples, we also add modified documents that both annotators tag as "non-contradictory," indicating such modification does not introduce document-level self-contradiction.

4 Evaluation

4.1 Evaluation Tasks and Metrics

We now describe the evaluation tasks and metrics for different experiments. We design three evaluation tasks, ranging from the simple "answer Yes or No" to the more complex "first judge, then give evidence". Our experiments and evaluation prompts are designed upon respective evaluation tasks.

4.1.1 Binary Judgment

Task The most straightforward way to evaluate the models is to test their abilities to distinguish between positive and negative examples. We do this by simply asking the model to provide a judgment on whether a document d is self-contradictory or not. In this setting, we evaluate the model on CONTRADOC .

[1] The annotators were native English speakers from the US with at least a Bachelor's degree in English.

Prompt Design We formalize this as the *Binary Judgment* task: Given a document, we ask the model if the document contains a self-contradiction. The model must answer with either "Yes" or "No".

Evaluation Metrics As CONTRADOC has balanced positive and negative cases, we use the standard Precision, Recall, F1 score, and Accuracy metrics to evaluate the models' binary judgment $j(d)$.

4.1.2 Self-Contradiction Top-k

Task In the zero-shot setting, the performance of the two aforementioned tasks can depend on how sensitive the model is to self-contradictions. If the model is under-sensitive, it might ignore non-critical self-contradictions; if it is over-sensitive, it might consider some minor potential inconsistencies in the document to be self-contradictory. Therefore, we design another task to find self-contradiction with top k evidence texts. While the self-contradiction introduced by our creation process is assumed to be the most obvious error in the document, it should appear within the top k evidence texts the model provides. Under this setting, the model is evaluated on CONTRADOC-POS .

Prompt Design We formalize this as the *Self-Contradiction Top-k*: Given a document with a self-contradiction, we ask the model to select the five most probable sentences that indicate the self-contradiction and rank them from high to low probability. We state in the prompt that the given document contains one self-contradiction.

Evaluation Metric Given the fact that a self-contradiction in the document is introduced by either inserting or replacing c_i or $c_i^{'}$ to the document, removing which would eliminate the self-contradiction in $\hat{d}_i$, thus we define c_i or $c_i^{'}$ as the oracle evidence e_i. Therefore, the evidences of self-contradiction given by the model must contain the corresponding e_i. Thus, we compare the evidences generated by the model with e_i using BertScore (Sun et al., 2022): if one of the evidences given by the model matches e_i with BertScore's Precision > 0.98 or Recall > 0.98, we consider it correct. To verify the evidences $E = \{s_j \mid j = 1, \ldots, k\}$ found by the model, the verification function $v(E)$ is given by:

$$v(E) = \begin{cases} \text{True} & \text{if } \exists s \in E \text{ such that} \\ & \max(\textsc{BertScore}(s, e_i)_{\text{Prec.}}, \\ & \quad \textsc{BertScore}(s, e_i)_{\text{Rec.}}) > 0.98 \\ \text{False} & \text{otherwise} \end{cases} \qquad (411)$$

Model	Accuracy	Precision	Recall	F1
GPT3.5	50.1%	100.0%	0.2%	0.4 %
GPT4	53.8%	97.0%	8.0%	15.6%
PaLM2	52.0%	61.0%	13.4%	22.0%
LLaMAv2	50.5%	51.0%	38.3%	43.7%

Table 1: Performance of different LLMs on **Binary Judgement** experiment.

Model	EHR ↑	Avg. Index (1-5) ↓
GPT3.5	42.8%	1.98
GPT4	**70.2%**	**1.79**
PaLM2	48.2%	2.36
LLaMAv2	20.4%	2.28

Table 2: Performance comparison of different LLMs on **Self-Contradiction in top-k** experiment. Evidence Hit Rate(EHR) by random sampling is 16%. Avg. Index (1-5) is the average index among the top-5 evidence texts where the self-contradiction was found.

We define *Evidence Hit Rate* (EHR) as the percentage of cases where the model could find the correct evidence. In practice, we choose $k = 5$ for top k. We calculate the EHR to represent the fraction of $v(E)$ = True for CONTRADOC-POS.

4.1.3 Judge then Find

Task Another drawback with Binary Judgment is that answering "Yes" does not necessarily mean the model can find the self-contradiction. Therefore, we design another task that requires not only binary judgment but also the evidence indicating the self-contradiction in the document, in case it answers "Yes" for the binary judgment task. In this setting, the model is evaluated on CONTRADOC .

Prompt Design We formalize the *Judge-then-Find* task as follows: Given a document, the model needs to determine whether the document has self-contradictions by answering "Yes" or "No." If the answer is Yes, the model also needs to provide supporting evidence by quoting sentences that can indicate the self-contradiction in the document.

Evaluation Metric In addition to the metrics mentioned in Section 4.1.1, an extra *Verification* $v(E^{'})$ is applied to the evidences $E^{'}$ provided by the model. Note that compared to E in *Self-Contradiction Top* k, $E^{'}$ usually contains a pair of evidence texts instead of k. The *Evidence Hit Rate* (EHR) here is defined as the percentage of cases where the model could find the correct evidence when it answered "Yes" wherever applicable. We measure EHR by automatically verifying the supporting evidence provided by the LLMs. It is evaluated only on TPs in this setting, and we show the real accuracy $R - acc(pos)$ over the positive subset CONTRADOC-POS to represent the fraction of $j(d) \wedge v(E^{'})$ = True.

The corresponding prompts for all three experimental settings are in Appendix E.

4.2 Automatic Evaluation results

Table 1 shows the results for the *Binary Judgment* Task. We find that all models struggle with detecting self-contradictory documents and predict "No" for most documents, as shown by the low recall values. We observe that LLaMAv2 achieves higher numbers only because it tends to predict "Yes" while other models tend to predict "No" for most of the cases. The accuracy on the entire dataset, i.e., CONTRADOC-POS and CONTRADOC-NEG, is around 50%, suggesting that the models have a near-random performance.

Table 2 shows the results for the *Self-Contradiction Top-k* Task, where, given a self-contradictory document, the models need to refer to the top-5 probable sentences that can imply the self-contradiction. We find that GPT4 outperforms the other models by a big margin and can correctly detect 70% of self-contradictions. PaLM2 is better than GPT3.5 and can correctly detect self-contradictions in 48% of the documents compared to 43%. Finally, LLaMAv2 performs the worst and can detect self-contradictions in only 20% of the documents. We also find that, on average, GPT4 can find the evidence at the 1.79th position out of 5, showing that it is not only best at finding the evidence sentences but also prioritizing them. Note that for all models, the average index that the evidence is found < 3, which indicates that the models do rank the evidence by probability of self-contradiction. We also provide a deeper analysis in Section 4.4.

Finally, Table 3 shows the results for the *Judge then Find* experiment. In the first part of the task, i.e., answering if the document is self-contradictory or not, similar to results in Table 1, we find that PaLM2 and LLaMAv2 have a greater bias to answer "Yes", compared to the GPT models. This is seen in the high TP and FP rates of the two models. However, the low Evidence Success Rates indicate

Models	Precision	Recall	F1 Score	TP rate	FP rate	TN rate	FN rate	Evidence Hit Rate	R-acc(pos)
GPT3.5	57.0%	62.0%	41.0%	20.6%	12.8%	36.9%	29.7%	41.0%	16.8%
GPT4	88.0%	39%	54.0%	19.6%	2.7%	46.2%	31.5%	92.7%	35.6%
PaLM2	52.0%	83.0%	64.0%	41.5%	37.6%	12.0%	9.0%	41.0%	33.7%
LLaMAv2	50.0%	95.0%	65.0%	48.0%	48.6%	1.12%	2.3%	14.5%	13.8%

Table 3: Performance comparison of different LLMs on *Judge then Find* experimental setting. **Precision, Recall, F1** and **TP, FP, TN, and FN** rates are calculated on the entire dataset before verification, i.e., on "Yes/No" prediction. **Evidence Hit Rate** is the percentage of cases where the model could find the correct evidence when it answered "Yes". **R-acc(pos)** denotes the fraction of positive data points confirmed by 'yes' judgments and evidence hits.

Models	**TP** rate	**FP** rate	**TN** rate	**FN** rate	Evidence Hit Rate	R-acc(pos)
Human	18.0%	6.7%	43.3%	32.0%	74.1%	26.7%
GPT3.5	20.7%	15.3%	34.7%	29.3%	25.8%	10.7%
GPT4	20.0%	4.7%	45.3%	30.7%	86.7%	34.7%

Table 4: Performance comparison of humans and different LLMs on *Judge then Find* experimental setting on a subset containing 75 positive documents and 75 negative documents. The metrics are similar to those in Table 3.

that the models fail to locate the correct evidence when they answer "Yes" to a self-contradictory document. LLaMAv2, in particular, can only find the correct evidence 14.5% of the time, while GPT3.5 and PaLM2 find correct evidence 41% of the time. Even though GPT4 might only be able to find 19.6% of the CONTRADOC-POS, it can provide the correct evidence for 92.7% of them. GPT4 performs the best in terms of real accuracy, followed closely by the PaLM2 model. In summary, we present the following key observations:

- GPT4 performs the best overall, whereas LLaMAv2 performs the worst.
- PaLM2 and LLaMAv2 are biased to answer Yes more often on yes/no prompts, whereas GPTs provide a more balanced output. However, all four models struggle with the yes/no prompts.
- While GPT4 predicts "yes" less than other models, the evidence hit rate of GPT4 is significantly higher than others, which shows that it is conservative and only answers "yes" when being certain about the self-contradiction.

Categories	Attributes	# docs	GPT3.5	GPT4
Overall	-	449	42.8%	70.2%
Document Type	news	158	45.6%	65.8%
	wiki	150	48.0%	82.0%
	story	141	34.0%	62.4%
Document Length	100-500	50	50.0%	64.0%
	500-1000	184	40.2%	69.6%
	1000-1500	143	44.1%	74.1%
	1500-2200	72	41.7%	68.1%
Self-Contra Scope	global	155	51.0%	89.0%
	local	220	38.6%	63.2%
	intra	73	37.0%	50.7%
Self-Contra Type	Negation	87	56.3%	85.1%
	Numeric	65	58.5%	87.7%
	Content	288	43.4%	74.7%
	P/V/O	101	25.7%	61.4%
	E/M/F	86	29.1%*	50.0%
	Factual	54	40.7%	66.7%
	Relation	25	40.0%	72.0%
	Causal	36	33.3%	55.6%

Table 5: Fine-grained performance of different LLMs on top-k judgment. The scores denote the Evidence Hit Rate. Numbers marked with an asterisk (*) denote Evidence Hit Rate is not statistically significant against random with p-value > 0.05. P/V/O refers to Perspective/View/Opinion while E/M/F refers to Emotion/Mood/Feeling.

4.3 Human Performance

We construct a balanced set of documents from our dataset with 150 documents in total and evaluate humans'performance on the *Judge then Find* task. Each document is evaluated by one annotator[2]. We then also compare their performance with the performance of GPT3.5 and GPT4 on the same documents. Table 4 shows the performance comparison. We use the same metrics as the *Judge then Find* experimental setting.

We find that overall, humans perform better than GPT3.5 but not GPT4. Specifically, we find that humans are the worst at finding TP cases. However,

[2]The annotators for this task are different from those who worked to verify documents before

they are much better than GPT3.5 at finding the self-contradiction evidence and does not point out false self-contradiction.

A possible reason for humans' poor performance is that humans might fail to keep track of details when the document is long, making them miss some self-contradictions. This is a different setting from the annotator verification process, where two potentially contradictory sentences are highlighted, which makes the task easier for humans.

4.4 Ablation Study

We now discuss the fine-grained analysis of various models' outputs to get a deeper understanding of their performance on the task of self-contradiction detection and answer the research questions mentioned in Section 3.1. We choose the model outputs of GPT3.5 and GPT4 from the **Self-Contradiction Top-k** experimental setting for this analysis. We use the probability (p-value) of finding equivalent successes in a binomial test to show the statistical significance of the results against random selecting k sentences from the document. Table 5 shows the EHR of these models in detecting the self-contradictory statement given in the document.

RQ1 Among the three document types, we find that models have the highest EHR on Wikipedia documents, followed by News and Stories. GPT4 can detect the self-contradictory statements in 82% of the Wikipedia documents, compared to 48% of the cases for GPT3.5. For Stories, the evidence hit rate of GPT4 and GPT3.5 drops to 62.4% and 34.04%, respectively.

RQ2 For both GPT3.5 and GPT4, there is no significant drop in EHR as the document length increases or the other way around. This suggests that the document length is not the main factor determining model's ability to detect self-contradictions. However, documents with relatively short lengths (100-500 tokens) are easier for GPT3.5 to detect the self-contradiction within.

RQ3 We find that for both GPT3.5 and GPT4, "global" self-contradictory documents had a higher EHR than "local" and "intra". This is in contradiction to our hypothesis that self-contradiction with evidence texts far away might be harder. This can be due to label dependencies shown in Figure 2 (discussed ahead).

RQ4 As we consider the types of self-contradiction types, we find that more objective self-contradiction types, like Numeric and Negation, are the easiest to detect, while more subjective ones like Emotion/Mood/Feeling and Perspective/View/Opinion are hard. We argue this might be because LLMs are pre-trained on more fact-checking tasks aiming to verify facts compared to emotion-consistency tasks.

Dataset Label Dependencies The fine-grained results in Table 5 can also be attributed to the label dependencies shown in Figure 2. As mentioned before, Wikipedia documents are more likely to contain Negation, Numeric and Factual self-contradiction, whereas Stories are more likely to contain Emotion/Mood/Feeling and Perspective/View/Opinion self-contradictions. Similarly, the performance differences in different scopes(global/local/intra) might also be attributed to their distributions of contradiction types. Here, we argue that the models' performance is more related to the self-contradiction type instead of where the self-contradiction is presented or the type of the document.

We also conduct additional experiments on the effects of prompt formatting and finding contradictory sentences in the document in Appendix F. We also found that our proposed task cannot be easily tackled by adapting NLI to a pair-wise setting.

5 Conclusion

In this work, we present one of the first steps in investigating the task of document-level self-contradictions. We create ContraDoc, a well-annotated dataset for this task, which contains 449 self-contradictory documents spanning over three domains and containing multiple types of self-contradictions. The dataset is annotated by humans and contains information about the scope and type of self-contradiction as well as the evidence to detect self-contradictions. We then investigate the capabilities of four state-of-the-art LLMs, namely, GPT3.5, GPT4, PaLM2, and LLaMAv2, on this dataset. We find that overall, GPT4 performs the best and even outperforms humans on the task. However, we also find that there is still a long way to go before GPT4 can reliably detect self-contradictions. We release this dataset and all the associated code for the community to use and develop better document-level reasoning capabilities in LLMs. As part of future work, we plan to investigate the capabilities of LLMs to fix the self-contradictions in the documents.

Limitations

Our aim was to create a dataset of self-contradictory documents that sound natural. However, as all self-contradictions are created and inserted automatically, the self-contradictory documents do not always mimic how humans make mistakes or introduce self-contradictions, even though we use humans-in-the-loop. Another limitation is that for some self-contradiction types, we only collected limited data points; for example, there are only 25 documents with *Relation* self-contradictory type in our dataset. Finally, in this work, we only study self-contradictions in English, and our dataset contains documents that are written in English.

Ethics Impact

We propose ContraDoc to encourage attention to the task of self-contradiction, a crucial area that has been notably overlooked in previous research. This task holds substantial practical value in real-world applications like document understanding, evaluation and quality. Moreover, this task has potential applications in legal and academic document analysis, where identifying contradictions can be critical. It's important to clarify that our goal is to augment the capabilities of human professionals, not to replace them. We propose an annotated dataset with automatic evaluation metrics can be a valuable asset to the NLP community, enabling the development and testing of new AI algorithms in this space. Since we build upon fully open-source datasets, we do not see it having any potential risks or negative ethical issues.

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020b. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow. 2003. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 38–45.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, page 177–190, Berlin, Heidelberg. Springer-Verlag.

Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. 2008. Finding contradictions in text. In *Proceedings of ACL-08: HLT*, pages 1039–1047, Columbus, Ohio. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. 2023. Is GPT-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11173–11195, Toronto, Canada. Association for Computational Linguistics.

Etienne Fortier-Dubois and Domenic Rosati. 2023. Using contradictions improves question answering systems. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 827–840, Toronto, Canada. Association for Computational Linguistics.

Arthur C Graesser and Cathy L McMahen. 1993. Anomalous information triggers questions when adults solve quantitative problems and comprehend stories. *Journal of Educational Psychology*, 85(1):136.

Sanda Harabagiu, Andrew Hickl, and Finley Lacatusu. 2006. Negation, contrast and contradiction in text processing. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, page 755–762. AAAI Press.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Cheng Hsu, Cheng-Te Li, Diego Saez-Trumper, and Yi-Zhan Hsu. 2021. Wikicontradiction: Detecting self-contradiction articles on wikipedia. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 427–436.

Di Jin, Sijia Liu, Yang Liu, and Dilek Hakkani-Tur. 2022. Improving bot response contradiction detection via utterance rewriting. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 605–614, Edinburgh, UK. Association for Computational Linguistics.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Puneet Mathur, Gautam Kunapuli, Riyaz Bhat, Manish Shrivastava, Dinesh Manocha, and Maneesh Singh. 2022. DocInfer: Document-level natural language inference using optimal evidence selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 809–824, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation.

Yixin Nie, Mary Williamson, Mohit Bansal, Douwe Kiela, and Jason Weston. 2021. I like fish, especially dolphins: Addressing contradictions in dialogue modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1699–1713, Online. Association for Computational Linguistics.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. OpenAI Blog.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

José Otero and Walter Kintsch. 1992. Failures to detect contradictions in a text: What readers believe versus what they read. *Psychological Science*, 3(4):229–236.

Nicholas Pangakis, Samuel Wolken, and Neil Fasching. 2023. Automated annotation with generative ai requires validation. *ArXiv*, abs/2306.00176.

Tal Schuster, Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, and Donald Metzler. 2022a. Stretching sentence-pair nli models to reason over long documents and clusters.

Tal Schuster, Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, and Donald Metzler. 2022b. Stretching sentence-pair NLI models to reason over long documents and clusters. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 394–412, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, Mike Schaekermann, Amy Wang, Mohamed Amin, Sami Lachgar, Philip Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Nenad Tomasev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle Barral, Dale Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. 2023. Towards expert-level medical question answering with large language models.

Tianxiang Sun, Junliang He, Xipeng Qiu, and Xuanjing Huang. 2022. BERTScore is unfair: On social bias in language model-based metrics for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3726–3739, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-level machine translation with large language models.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? gpt-3 can help. *ArXiv*, abs/2108.13487.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021a. Docnli: A large-scale dataset for document-level natural language inference.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021b. DocNLI: A large-scale dataset for document-level natural language inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online. Association for Computational Linguistics.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2023. Benchmarking large language models for news summarization.

Chujie Zheng, Jinfeng Zhou, Yinhe Zheng, Libiao Peng, Zhen Guo, Wenquan Wu, Zheng-Yu Niu, Hua Wu, and Minlie Huang. 2022. CDConv: A benchmark for contradiction detection in Chinese conversations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 18–29, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Dataset Details

We use three publically available datasets covering different domains to build CONTRADOC. More specifically, we use the following datasets:

- **News Articles**: CNN-DailyMail dataset (Hermann et al., 2015), an open-source corpus of 93k articles from CNN and 220k articles from Daily Mail and collect 158 documents for CONTRADOC-POS.
- **Stories**: NarrativeQA (Kočiskỳ et al., 2018), which is an open-source question-answering dataset and consists of 1,572 stories and their human-generated summaries. We collected 141 summaries for CONTRADOC-POS.
- **Wikipedia**: WikiText (Merity et al., 2016), an open-source language modelling dataset containing verified Wikipedia documents and select 150 documents for CONTRADOC-POS .

We release our dataset under Apache 2.0 license [3].

B Model details

We use the following state-of-the-art LLMs to test both open-source and closed-source models in a zero-shot setting on CONTRADOC .

- **GPT3.5**: Also called ChatGPT[4], this is an improved version of GPT3 (Brown et al., 2020b) optimized for chat. We use the *gpt-3.5-turbo-0613* model from the OpenAI API[5].
- **GPT4** (OpenAI, 2023): GPT4 is the latest iteration of the GPT models and is also optimized for chat. We use the *gpt-4-0613* model from the OpenAI API.
- **PaLM2** (Anil et al., 2023): We use the PaLM 2 model (*text-bison*) from the Vertex AI platform from Google Cloud[6].
- **LLaMAv2** (Touvron et al., 2023): We use the *Llama-2-Chat-70B* model for our experiments. We used the best performing model that is fine-tuned on dialog data to follow 0-shot instruction.

[3] https://www.apache.org/licenses/LICENSE-2.0
[4] https://openai.com/blog/chatgpt
[5] https://api.openai.com/
[6] https://cloud.google.com/vertex-ai/docs/generative-ai/learn/models

Unless otherwise specified, we use the default configurations and decoding parameters for all our experiments.

C Types of self-contradictions

ContraDoc contains eight types of self-contradictions. Table 6 provides the definitions for each self-contradiction type and example transformation of a sentence. This information was used by our annotators for evaluating and creating the dataset.

D Questions for Annotation

Annotators, guided by comprehensive guidelines, were tasked to answer the following questions:

Q1. Do you think the two statements contradict each other?

Q2. (If applicable): Is the position of the inserted statement (red color) feasible?

Q3. Overall, do you think it makes an acceptable contradictory document?

Q4. How close in the context of the modified sentence can you find the evidence for the self-contradiction? (As described in 3.1)

Q5. Select Type(s) of self-contradiction.

E Prompts for experiment setting

For evaluating the different LLMs on ContraDoc , we set up three experiments. Here, we provide the corresponding prompts for each of the experimental settings.

- **Binary Judgment Prompt**

  [Insert Document here]

  Determine whether the given document contains any self-contradictions. Only answer "yes" or "no"!

- **Self-Contradiction in Top k Prompt:**

  Self-Contradictory Article: An article is deemed self-contradictory when it contains one(self-conflict mention) or more statements that conflict with each other, making them mutually exclusive. The following article contains one self-contradiction. The task is to find where it is. Provide evidence by quoting mutually contradictory sentences from the article. Article:

  [Insert Document here]

  Please respond by giving the five most likely sentences that can reflect article-level contradiction(s), ranked by high to low possibility. Don't explain.

- **Judgment then Find Prompt:**

  The task is to determine whether the article contains any self-contradictions. If yes, provide evidence by quoting mutually contradictory sentences in a list of strings in Python. If no, give an empty list.

  [Insert Document here]

  Response: Form your answer in the following format (OR options are provided):

  Judgment: yes OR no

  Evidence: ["sentence1", "sentence2", ..., "sentenceN"] OR []

- **Prompt for Effect of Prompts experiment:**

  Go over the following document and check if there is any self-contradiction (e.g., conflict facts) in it? If there are issues related to consistency or coherence, please also point them out.

F Additional Sensitivity Analysis

Effect of Prompts Since we enforce model outputs to a fixed format, this might negatively affect the model performance. This is more true for GPT3.5 than GPT4, which has better instruction-following capability. Thus, for 75 documents with self-contradictions, we ask GPT3.5 to generate predictions without putting constraints on the output format (prompt in Appendix E) and ask humans to evaluate the responses. For 26.4% cases, it answers "No"; for 45.8% of the cases, it provides incorrect evidence; only for 27.8% of the cases is it able to find the correct evidence (alongside other incorrect evidence). This suggests that the model performance is still far from satisfactory.

Detecting self-contradictory sentence Since we observe that models find it hard to find contradictions in a document, we evaluate the model's capability on an easier task to find a statement that directly contradicts a given sentence. Since our dataset contains documents that contain a pair of contradictory sentences, we provide the evidence sentence to the model and ask it to find the contradictory sentence in the document. GPT3.5 can detect 51.6% of the cases, while GPT4 can detect

Type	Definition	Original Statement	Generated Self-Contradiction
Negation	Negating the original sentence	Zully donated her kidney.	Zully never donated her kidney.
Numeric	Number mismatch or number out of scope.	All the donors are between 20 to 45 years old.	Lisa, who donates her kidney, she is 70 years old.
Content	Changing one/multiple attributes of an event or entity	Zully Broussard donated her kidney to a stranger.	Zully Broussard donated her kidney to her close friend.
Perspective / View / Opinion	Inconsistency in one's attitude/ perspective/opinion	The doctor spoke highly of the project and called it "a breakthrough"	The doctor disliked the project, saying it had no impact at all.
Emotion / Mood / Feeling	Inconsistency in one's attitude/ emotion/mood	The rescue team searched for the boy worriedly.	The rescue team searched for the boy happily.
Relation	Description of two mutually exclusive relations between entities.	Jane and Tom are a married couple.	Jane is Tom's sister.
Factual	Need external world knowledge to confirm the contradiction.	The road T51 was located in New York.	The road T51 was located in California.
Causal	The effect does not match the cause.	I slam the door.	After I do that, the door opens.

Table 6: Definition and example of sentence transformations for different types of self-contradictions.

77.2% of them. Such results suggest that LLMs do reasonably well in document-level contradiction detection if the exact sentence with contradiction is pointed out but not so otherwise, but perform much worse in finding self-contradiction if the exact sentence isn't pointed out for its reference.